BioInfoSummer 2004

Assignment 3 (Analysis of Microarray Data)

John Maindonald

Centre for Bioinformation Science, Australian National University

1 Preparation

Background Documents

The document **marray-wkshp04.pdf** is a revised version of the workshop notes. Be sure to have this available as you work through the exercises. Note also the document **marray-wkshp04-add.pdf**, which shows how to obtain plots more compactly, up to six to a page

Installation of the R software

Requirements are:

(a) R (2.0.0 or later), available from the Comprehensive R Archive Network (CRAN). Versions are available for Unix, Linux, Windows and Macintosh OS X.

(b) The *limma* and (as one source of data) *marray* packages for R, available from the Bioconductor web site.

(c) Either the *marray* package for R, or the separate files **SwirlSample.txt**, **fish.gal**, **swirl.1.spot**, **swirl.2.spot**, **swirl.3.spot** and **swirl.4.spot**, available from my web page `http://wwwmaths.anu.edu.au/~johnm/r/biosummer/swirldata`

(d) The function `slide.image()`, available from my web page `http://wwwmaths.anu.edu.au/~johnm/r/biosummer/r-functions` Download **slide.image.RData** and, optionally, **plotprintseq.RData** and **sixplot.RData**. Alternatively, any or all of the functions in these files can be loaded directly from my web page, using R's `loadURL()` function. Assuming a live internet connection, use the syntax:

```
loadURL("http://wwwmaths.anu.edu.au/~johnm/r/biosummer/r-functions/slide.image.RData")
```

Installation of R (Windows)

Users of non-Windows systems can find installation details on the web sites that are noted. The steps for Windows users are:

(a) If this is not otherwise available (e.g., from one of the CDs that I am making available), download the file **rw2001pat.exe** (~23MB) by going to the link `http://mirror.aarnet.edu.au/pub/CRAN/bin/windows/base/README.rw2001pat` (This is slightly preferable to **rw2001.exe**; some minor bugs have been fixed.)

(b) Click on the file **rw2001pat.exe** to start installation, and follow the instructions. You will be asked which components you wish to install. You must install the "Main files" (16.7MB).

It may is desirable to install, also, the Compiled HTML Help Files (this allows access to the help files through a browser). The Support Files for library(tcltk) (5.2MB) provide routines for building graphical user interfaces. These are not relevant to the present exercise.

(c) Right-click on the R icon that was created on the desktop, and click on Properties. The directory whose name appears in Start in will be the working directory that will, by default, be used for your work. A better choice than the default might be, e.g., **c:\r\biosummer\**.

NB: For using **c:\r\biosummer\** as the working directory, it must first be created.

Installation of Relevant BioConductor Packages

The packages that will be needed here are *limma* and (as a source for the data that is alternative to getting the data from the CD or from the CBIS web site) *marray*.

Start up R. Make sure that the computer is connected to the internet. On the R command line, type in each of the following in turn.

```
source("http://www.bioconductor.org/getBioC.R")
getBioC(libName="limma", bundle=FALSE)
getBioC(libName="marray", bundle=FALSE)
```

(Following each line, there will be a delay until the task is complete and R is ready for the next line.)

Linux, Unix and Mac users should be able to use this same approach.

Alternative 1: An alternative is to copy down the packages or copy them from a CD that includes them, then install them. To copy them from the web, go to `http://www.bioconductor.org`, click on Release 1.5 Packages, click on marray and download the Win32 version of that. (A right click may be necessary, allowing you to control the directory to which they are saved, and to prevent expansion of the zip files.) Click the Packages menu item, then on Install Package(s) from local zip file, etc.

Alternative 2: EITHER click Packages, then Install Package(s) from Bioconductor, etc.
[On the Mac OS X Aqua GUI for R, click on Packages & Data, then Package Installer, set the repository to BioConductor (binaries), click on Get List, etc.]
OR click the Packages menu item, then Install Package(s) from local zip files..., etc.

Alternative 3: Some may wish to install one or more complete bundles of packages. For example, use the following to install the *exprs* bundle, which includes *limma*

```
getBioC(libName="exprs", relLevel="release")
```

To get further details of the bundles that are available, look about two thirds of the way down the web page
`http://www.bioconductor.org/faq.html`

Linux, Unix and Mac users should be able to use this same approach.

Data from the Swirl Experiment

To get a description of the experimental work that generated these data, type on to the R command line

```
library(marray)
help(swirl)
```

First, find where the files are located. Type in

```
.Library
```

The files that are needed are in the **marray/swirldata** subdirectory of this path.

The files are **swirl.1.spot**, **swirl.3.spot**, **swirl.2.spot**, **swirl.4.spot**, **fish.gal**, and **SwirlSample.txt**. Use the GUI or a text window command line to copy these files to the R Start In directory. If the suggestion made above was followed, the Start In directory will be **c:\r\biosummer\**

Reading Data into R

Go back to the R command line. The first command below gets the file names. It finds all the files in the working directory that have names that end in "**.spot**".

```
library(limma)
targets <- readTargets("SwirlSample.txt")
```

Next, use the function `read.maimages()` to read in the data, the function `readGAL()` to attach annotation information, and the function `getLayout()` to attach information on the slide layout. Both functions are from the *limma* package.

```
swirlRG <- read.maimages(targets$Names, source = "spot")
swirlRG$genes <- readGAL("fish.gal")
swirlRG$printer <- getLayout(swirlRG$genes)
names(swirlRG)
print(swirlRG$targets)   # Examine the information stored here
head(swirlRG$genes)      # Check the first few rows only
  # Proceed similarly, if required, for other named elements
```

2 The Assignment Starts Here!

2.1 Exercise 1 – Names

Explain the different names that were obtained upon typing `names(swirlRG)`.

2.2 Exercise 2 – Spatial Plots

The following gives a spatial plot for the 10% of values (5% at each extreme) of the red (Cy5) signal that are most extreme, for the first slide:

```
slide.image(swirlRG$R[, 1], layout = swirlRG$printer)
```

> Alternatively, copy down the file **slide.image.RData** from the internet, place it in your working directory, and type:
>
> ```
> source("slide.image.RData")
> ```
>
> This will load the R function `slide.image()` into your R session. Then proceed with
>
> ```
> slide.image(swirlRG$R[, 1], layout = swirlRG$printer)
> ```

Repeat the above for the red signal and for the green signal, for each of the slides, and comment on the results. Is there any spatial pattern to what you see. How might any such pattern be explained?

2.3 Exercise 3 – MA plots & Normalization

(a) First, note the pattern of logratios, before normalization. The smooth curves in the plot are the loess curves that will be used for normalization.

```
swirlMAn <- normalizeWithinArrays(swirlRG, method = "none")
plotPrintTipLoess(swirlMAn)
rm(swirlMAn)
```

(b) Apply loess normalization, and check the MA plots:

```
swirlMAw <- normalizeWithinArrays(swirlRG)
plotPrintTipLoess(swirlMAw)
```

(c) Check whether normalization seems required between arrays:

```
boxplot(swirlMAw$M ~ col(swirlMAw$M), names = colnames(swirlMAw$M))
```

(d) Scale normalize between arrays, and repeat the boxplot:

```
swirlMA <- normalizeBetweenArrays(swirlMAw)
rm(swirlMAw)
boxplot(swirlMA$M ~ col(swirlMA$M), names = colnames(swirlMA$M))
```

2.4 Exercise 4 – Checks for Differential Expression

(a) First, fit a simple statistical model that allows for the possible effect of the dye swap, and that can be used as the basis for checks for differential expression. The calculations require a design vector that has -1 wherever there is a dye swap:

```
design <- c(-1, 1, -1, 1)
swirlfit <- lmFit(swirlMA, design)
```

(b) From the fit object, calculate empirical Bayes moderated *t*-statistics.

```
efit <- eBayes(swirlfit)
## You might like to examine a qq-plot
qqt(efit$t, df = efit$df.prior + efit$df.residual, pch = 16,
    cex = 0.2)
```

(c) Print out a table that shows the top 25 differentially expressed genes, in order of the values of the moderated *t*-statistic:

```
options(digits = 3)
topTable(efit, number = 25, adjust="fdr")
```

(d) Repeat item (c) above, but now omitting `adjust="fdr"`. (There is now no adjustment for selection of the largest *t*-statistics, from a large number of observations.)

(e) Repeat item (c) above, but now set the prior estimate of the proportion to be 0.1. (For this, specify `eBayes(swirlfit, proportion=0.1, adjust="fdr")`.) How does this affect the number of genes for which the Bayes factor (B) is greater than 5?

3 Optional extra – not part of the assignment

Finally, here is another type of plot:

```
plot(efit$coef, efit$lods, pch = 16, cex = 0.2, xlab = "log(fold change)",
    ylab = "log(odds)")
ord <- order(efit$lods, decreasing = TRUE)
top8 <- ord[1:8]
text(efit$coef[top8], efit$lods[top8], labels = swirlRG$genes[top8,
    "Name"], cex = 0.8, col = "blue")
```